Synthesizing Switching Controllers for Hybrid Systems by Continuous Invariant Generation

Deepak Kapur[1], Naijun Zhan[2], and Hengjun Zhao[2,3]

[1] Dept. of Comput. Sci., University of New Mexico, Albuquerque, NM, USA
kapur@cs.unm.edu
[2] State Key Lab. of Comput. Sci., Institute of Software, CAS, Beijing, China
znj@ios.ac.cn
[3] University of Chinese Academy of Sciences, Beijing, China
zhaohj@ios.ac.cn

Abstract. We extend a template-based approach for synthesizing switching controllers for semi-algebraic hybrid systems, in which all expressions are polynomials. This is achieved by combining a QE (quantifier elimination)-based method for generating continuous invariants with a qualitative approach for predefining templates. Our synthesis method is relatively complete with regard to a given family of predefined templates. Using qualitative analysis, we discuss heuristics to reduce the numbers of parameters appearing in the templates. To avoid too much human interaction in choosing templates as well as the high computational complexity caused by QE, we further investigate applications of the SOS (sum-of-squares) relaxation approach and the template polyhedra approach in continuous invariant generation, which are both well supported by efficient numerical solvers.

1 Introduction

Hybrid systems, in which computations proceed both by continuous evolutions as well as discrete jumps simulating transition from one mode to another mode, are often used to model devices controlled by computers in many application domains [1]. Combining ideas from state machines in computer science and control theory, formal analysis, verification and synthesis of hybrid systems have been an important area of active research. In verification problems, a given hybrid system is required to satisfy a desired safety property e.g. that the temperature of a nuclear reactor will never go beyond a maximum threshold, as it may cause serious economic, human and/or environmental damage, thus implying that the system will never enter any unsafe state. A synthesis problem is harder given that the focus is on designing a controller that ensures the given system will satisfy a safety requirement, reach a given set of states, or meet an optimality criterion, or a desired combination of these requirements.

Automata-theoretic and logical approaches have been primarily used for verification and synthesis of hybrid systems [2,4,42]. In [4,42], a general framework for controller synthesis based on hybrid automata was proposed, which relies on *backward reachable set* computation and *fixed point iteration*. Two restrictions of this approach are (i) the computation of backward reachable set is hard for most continuous dynamics, and (ii) termination of the fixpoint iteration procedure cannot be guaranteed, even for those hybrid systems whose backward reachable sets are easily computable. Thus most of the research, e.g. [14], focuses on overcoming the above two restrictions.

Recently, a deductive approach for verification and synthesis based on constraint solving was proposed in [11,27,26,38,20,37]. The central idea is to reduce verification and synthesis problems of hybrid systems to invariant generation problems, much like verification of programs. As proposed in [16,15,31], if invariants are hypothesized to be of certain shapes, then corresponding templates with associated parameters can be used and the invariant generation problem can be reduced to constraint solving over parameters by quantifier elimination. This methodology is used in [38] for synthesizing switching controllers meeting safety requirements, while in [40], the approach is extended for satisfying both safety and reachability requirements. A common problem with template-based method is that it heavily relies on a user specifying the shape of invariants that are of interest, thus making it interactive and user driven, raising doubts about its scalability and automation. Besides, the inference rules for inductive invariants in [39,38,40] are sound and complete for classes of invariants, e.g. smooth, quadratic and convex invariants, but are not complete for generic semi-algebraic sets.

Inspired by [17,4,38] and [21], we extend in this paper the template-based invariant generation approach for synthesizing switching controllers of hybrid systems to meet given safety requirements. The paper makes the following contributions:

- We formalize the solution to switching controller synthesis problem (Problem 1) of hybrid systems in terms of continuous invariants (see Theorem 1), and thus lay the foundation of the synthesis method based on continuous invariant generation using templates and constraint solving.
- In the QE-based synthesis framework we use the method for continuous invariant generation proposed in [21] as an integral component. This method is proved in [21] to be sound and relatively complete with respect to a given shape of invariants (i.e. a given family of predefined templates). As a result, in contrast to the methods used in [39,38,40], there is more flexibility in our approach because of the possibility of discovering all possible invariants of the given shape.
- Using the qualitative approach proposed in [17] for analyzing continuous evolution in certain modes of a hybrid system, we can develop heuristics to determine a more precise shape of templates to be used as invariants, thus reducing the numbers of parameters appearing in templates.
- We further improve the degree of automation and scalability of the template-based method in two ways: (i) for general polynomial templates, using *sum-of-squares* (SOS) relaxation, the constraint on parameters appearing in templates is transformed into a *semi-definite program*(SDP), which is convex and

thus can be solved efficiently; (ii) for linear systems and a special type of templates—template polyhedra, again by sacrificing relative completeness, the continuous invariant generation problem can be reduced to a BMI (*bilinear matrix inequality*) feasibility problem, which is also much easier to solve (numerically) than QE.

Related Work

Our work in this paper resembles [38] but differs in that: i) our method is cast in the setting of hybrid automata and searches for a family of continuous invariants that refine the original domains, rather than a single global controlled invariant; ii) a sound and complete criterion is used in continuous invariant generation; iii) various techniques are applied for scalability.

The SOS relaxation approach has been successfully used in safety verification of hybrid systems. In [28,29], the authors used the SOSTOOLS software package [30] to compute *barrier certificates* for polynomial hybrid systems. In [19,45], the authors proposed a hybrid symbolic-numeric approach to compute exact inequality invariants of hybrid systems, by first solving (bilinear) SOS programming numerically and then applying *rational vector recovery* techniques.

A necessary and sufficient condition for positive invariance of convex polyhedra for linear continuous systems was provided in [7]. This condition is extended to linear systems with open polyhedral domain for our need in the paper. Template polyhedra was used in [33,32] to compute positive invariants of hybrid systems by *policy iteration*, which differs from our treatment of the problem using BMI. Recently, a method for computing polytopic invariants for polynomial dynamical systems using template polyhedra and linear programming was proposed [35].

Mathematical programming techniques and relevant numerical solvers have also been widely applied to static program analysis. Actually, the template polyhedra abstract domain was first proposed in [34] to generate linear program invariants using linear programming. In [8], to verify invariance and termination of semi-algebraic programs, verification conditions are abstracted into numerical constraints using Lagrangian relaxation or SOS relaxation, which are then resolved by efficient SDP solvers.

In our recent work [46], we studied an optimal switching controller synthesis problem arising from an industrial oil pump system with piece-wise constant continuous dynamics. A hybrid approach combining symbolic computation with numerical computation was developed to synthesize safe controllers with better optimal values.

Paper Structure. To be completed.
1
2
3
4

5
6
7

2 Problem Description

Following [4,42], we use hybrid automata to model hybrid systems.

Definition 1 (Hybrid Automaton). *A hybrid automaton (HA) is a system* $\mathcal{H} \mathrel{\widehat{=}} (Q, X, f, D, E, G)$*, where*

- $Q = \{q_1, \ldots, q_m\}$ *is a set of discrete states;*
- $X = \{x_1, \ldots, x_n\}$ *is a set of continuous state variables, with* $\mathbf{x} = (x_1, \ldots, x_n)$ *ranging over* $\mathbb{R}^n$*;*
- $f : Q \to (\mathbb{R}^n \to \mathbb{R}^n)$ *assigns to to each discrete state* $q \in Q$ *a vector field* $\mathbf{f}_q$*;*
- $D : Q \to 2^{\mathbb{R}^n}$ *assigns to each discrete state* $q \in Q$ *a domain* $D_q \subseteq \mathbb{R}^n$*;*
- $E \subseteq Q \times Q$ *is a set of discrete transitions;*
- $G : E \to 2^{\mathbb{R}^n}$ *assigns to each transition* $e \in E$ *a switching guard* $G_e \subseteq \mathbb{R}^n$ *.*

Remark 1. For ease of presentation, we make the following assumptions:

- for all $q \in Q$, $\mathbf{f}_q$ is *polynomial* vector function; besides, $\mathbf{f}_q$ defines a *complete* vector field, that is, for any $\mathbf{x}_0 \in \mathbb{R}^n$, the solution to the differential $\dot{\mathbf{x}} = \mathbf{f}_q$ uniquely exists on $[0, \infty)$;
- for all $q \in Q$ and all $e \in E$, D_q and G_e are *closed semi-algebraic* sets[1];
- the initial condition in each discrete mode is assumed to be identical with the domain, and all reset functions are assumed to be identity mappings.

We use a nuclear reactor system discussed in [3,12,17] as a running example through this paper.

Example 1. The nuclear reactor system consists of a reactor core and a cooling rod which is immersed into and removed out of the core periodically to keep the temperature of the core, denoted by x, in a certain range. Denote the fraction of the rod immersed into the reactor by p. Then the initial specification of this system can be represented using the hybrid automaton in Fig. 1.

The semantics of a hybrid automaton $\mathcal{H}$ can be defined by the set of trajectories it accepts. For the formal definitions of *hybrid time set* and *hybrid trajectory* the readers are referred to [42]. We denote the set of trajectories of $\mathcal{H}$ by $\mathcal{T}r(\mathcal{H})$, ranged over $\omega, \omega_1, \ldots$. All trajectories of $\mathcal{H}$ starting from the initial state $(q_0, \mathbf{x}_0)$ is denoted by $\mathcal{T}r(\mathcal{H})(q_0, \mathbf{x}_0)$.

The domain of a hybrid automaton $\mathcal{H}$ is defined as $D_{\mathcal{H}} \mathrel{\widehat{=}} \bigcup_{q \in Q}(\{q\} \times D_q)$. We call $\mathcal{H}$ *non-blocking* if for any $(q, \mathbf{x}) \in D_{\mathcal{H}}$, there is a hybrid trajectory from

[1] A set $A \subseteq \mathbb{R}^n$ is called semi-algebraic if there is a quantifier-free polynomial formula φ s.t. $A = \{\mathbf{x} \in \mathbb{R}^n \mid \varphi(\mathbf{x}) \text{ is true}\}$.

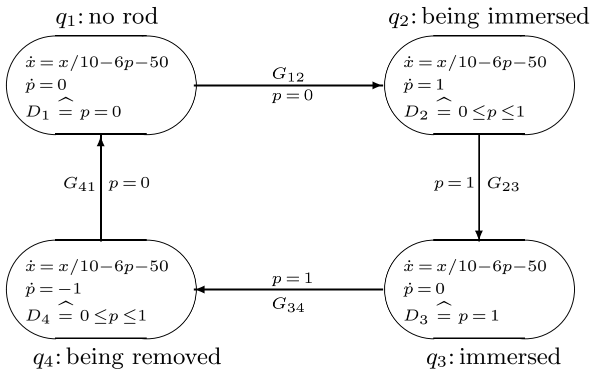

Fig. 1. Nuclear reactor temperature control.

$(q, \mathbf{x})$ which can either be extended to infinite time $t = \infty$ or execute infinitely many discrete transitions; otherwise $\mathcal{H}$ is called *blocking.*

A *safety requirement* S assigns to each mode $q \in Q$ a safe region $S_q \subseteq \mathbb{R}^n$, i.e. $S = \bigcup_{q\in Q}(\{q\} \times S_q)$. Alternatively, there could be a global safety requirement S which all modes are required to satisfy.

According to [4], the switching controller synthesis problem with regard to a given safety requirement can be formally defined as follows:

Problem 1 (Controller Synthesis for Safety). Given a hybrid automaton $\mathcal{H}$ and a safety property S, find a hybrid automaton $\mathcal{H}' = (Q, X, f, D', E, G')$ such that

(r1) Refinement: for any $q \in Q$, $D'_q \subseteq D_q$, and for any $e \in E$, $G'_e \subseteq G_e$;
(r2) Safety: for any trajectory ω that $\mathcal{H}'$ accepts, if $(q, \mathbf{x})$ is on ω, then $\mathbf{x} \in S_q$;
(r3) Non-blocking: $\mathcal{H}'$ is non-blocking.

If such $\mathcal{H}'$ exists, then $\mathcal{SC} \widehat{=} \{G'_e \subseteq \mathbb{R}^n \mid e \in E\}$ is a *switching controller* satisfying safety requirement S, and $D_{\mathcal{H}'} \widehat{=} \bigcup_{q\in Q}(\{q\} \times D'_q)$ is the *controlled invariant set* rendered by $\mathcal{SC}$.

3 A QE-Based Approach

3.1 Continuous Invariant

Along the line of [38], we consider the switching controller synthesis problem by combining a relative complete method for generating continuous invariants in [21], and heuristics for predefining templates for these invariants using qualitative analysis in [17]. Below, we review the concept of *continuous invariant* used in [21] based on a related concept in [26].

Definition 2 (Continuous Invariant (CI)). *Given a mode $q \in Q$ in a hybrid automaton $\mathcal{H}$, a set $P \subseteq \mathbb{R}^n$ is called a continuous invariant of $(D_q, \mathbf{f}_q)$ if for* *all* $\mathbf{x}_0 \in P \cap D_q$ *and all* $T \geq 0$*, the solution* $\mathbf{x}(t)$[2] *of* $\dot{\mathbf{x}} = \mathbf{f}_q(\mathbf{x})$ *over* $[0, T]$ *with* $\mathbf{x}(0) = \mathbf{x}_0$ *satisfies*

$$(\forall t \in [0, T].\, \mathbf{x}(t) \in D_q) \longrightarrow (\forall t \in [0, T].\, \mathbf{x}(t) \in P) \ .$$

Intuitively, P is a CI of $(D_q, \mathbf{f}_q)$ if any continuous evolution starting from the intersection of P and D_q stays in P as long as it is still in D_q. If $D_q = \mathbb{R}^n$, then a CI of $(D_q, \mathbf{f}_q)$ coincides with the standard *(positive) invariant set* (see [5]) of the dynamical system defined by $\mathbf{f}_q$; otherwise if D_q is a proper subset of $\mathbb{R}^n$, then generally the notion of CI is weaker.

Example 2. Suppose $D_q \mathrel{\widehat{=}} x > 0$ and $\mathbf{f}_q = (-y, x)$. Obviously $P \mathrel{\widehat{=}} y \geq 0$ is not a positive invariant set of $\mathbf{f}_q$, whereas P is a CI of $(D_q, \mathbf{f}_q)$ according to Definition 2. See Fig. 2 for an illustration.

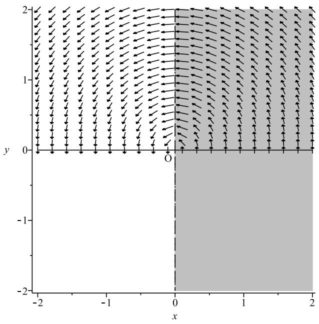

Fig. 2. Illustration of continuous invariant.

3.2 The Abstract Synthesis Procedure

To solve Problem 1 amounts to refining domains and guards of $\mathcal{H}$ by removing *bad* states from domain $D_{\mathcal{H}}$. A state $(q, \mathbf{x}) \in D_{\mathcal{H}}$ is *bad* if the hybrid trajectory starting from $(q, \mathbf{x})$ either blocks $\mathcal{H}$ or violates S; otherwise, it is called a *good* state. From Definition 2, we observe that the set of good states of $\mathcal{H}$ can be approximated using continuous invariants, which results in the following solution to Problem 1.

Theorem 1. *Let* $\mathcal{H}$ *and* S *be the same as in Problem 1. Suppose* D'_q *is a closed subset of* $\mathbb{R}^n$ *for all* $q \in Q$ *and* $D'_q \neq \emptyset$ *for at least one* q*. If we have*

(c1) for all $q \in Q$, $D'_q \subseteq D_q \cap S_q$*;*

[2] We assume the existence and uniqueness of solutions is guaranteed by $\mathbf{f}_q$.

(c2) for all $q \in Q$, D'_q is a continuous invariant of $(H_q, \mathbf{f}_q)$ with

$$H_q \mathrel{\widehat{=}} \Big(\bigcup_{e=(q,q')\in E} G'_e \Big)^c,$$

where $G'_e \mathrel{\widehat{=}} G_e \cap D'_{q'}$ and A^c denotes the complement of A in $\mathbb{R}^n$, then the HA $\mathcal{H}' = (Q, X, f, D', E, G')$ is a solution to Problem 1.

Proof. Please refer to the Appendix A. □

Remark 2. Intuitively, by (c1), D'_q is a refinement of D_q and is also contained in the safe region; by (c2), any trajectory starting from D'_q will either stay in D'_q forever, or finally intersect one of the transition guards enabling jumps from q to a certain q', thus guaranteeing satisfaction of the non-blocking requirement.

Based on Theorem 1, we give below the steps of a template-based method for synthesizing a switching controller.

(s1) **Template assignment:** assign to each $q \in Q$ a template parametrically specifying D'_q, which can be seen as a refinement of D_q and will be instantiated to be the continuous invariant at q;
(s2) **Guard refinement:** refine guard G_e for each $e = (q, q') \in E$ by setting $G'_e \mathrel{\widehat{=}} G_e \cap D'_{q'}$;
(s3) **Deriving synthesis conditions:** encode (c1) and (c2) in Theorem 1 into constraints on parameters appearing in templates;
(s4) **Constraint solving:** solve the constraints derived from (s3) in terms of the parameters;
(s5) **Parameters instantiation:** find an appropriate instantiation of D'_q and G'_e such that D'_q are closed sets for all $q \in Q$, and D'_q is nonempty for at least one $q \in Q$; if such an instantiation is not found, we choose a new set of templates and go back to (s1).

Remarks:

1. The implementability of the above method depends on the language used to specify the hybrid system, the safety property, as well as the templates chosen for their refinements. If all appearing expressions are specified using polynomials, the computability of the abstract procedure is guaranteed by Tarski's result [41]. This will be assumed in the rest of this paper.
2. In (s3), condition (c1) can be encoded into a first-order polynomial formula straightforwardly; encoding of (c2) into first-order polynomial constraints is based on our previous work in [21] about a relatively complete method for generating CIs (see Section 3.3).
3. We use *quantifier elimination* (QE) to solve the first-order polynomial constraints obtained in (s4).
4. The shape of chosen templates in (s1) determines the likelihood of success of the above procedure, as well as the complexity of QE in (s4). In Section 3.4, we discuss heuristics for choosing appropriate templates using the *qualitative analysis* discussed in [17].

3.3 A Relatively Complete Method for Generating CIs

In [21] we presented a sound and relatively complete approach for generating *semi-algebraic* CIs for $(D_q, \mathbf{f}_q)$ with semi-algebraic D_q and polynomial vector function $\mathbf{f}_q$. We review here the key ideas; for details the reader can consult [21]. Below, we drop the subscript corresponding to the mode q.

The basic idea can be explained as follows for the simplest case, namely $D \mathrel{\widehat{=}} h(\mathbf{x}) > 0$ and $P \mathrel{\widehat{=}} p(\mathbf{x}) \geq 0$. Let $\partial P \mathrel{\widehat{=}} p(\mathbf{x}) = 0$ be the boundary of P and $\mathbf{x}(t)$ be the continuous evolution of $\mathbf{f}$ starting from $\mathbf{x}_0$. It can be shown that P is a CI of $(D, \mathbf{f})$ if and only if for all $\mathbf{x}_0 \in \partial P \cap D$:

$$\exists \varepsilon > 0\, \forall t \in [0, \varepsilon].\, p(\mathbf{x}(t)) \in P \ . \tag{1}$$

Intuitively, (1) means that trajectories starting at the the boundary of P will stay in P for a small amount of time.

Given that p and $\mathbf{f}$ are polynomials and thus analytic, the *Taylor expansion* of $p(\mathbf{x}(t))$ at $t = 0$

$$p(\mathbf{x}(t)) = p(\mathbf{x}_0) + \frac{\mathrm{d}p}{\mathrm{d}t} \cdot t + \frac{\mathrm{d}^2 p}{\mathrm{d}t^2} \cdot \frac{t^2}{2!} + \cdots \tag{2}$$

converges in a neighborhood of 0.

Define the *Lie derivatives* of p along $\mathbf{f}$, $L_{\mathbf{f}}^n p : \mathbb{R}^n \longrightarrow \mathbb{R}$ for $n \in \mathbb{N}$, as follows:

- $L_{\mathbf{f}}^0 p(\mathbf{x}) = p(\mathbf{x})$;
- $L_{\mathbf{f}}^n p(\mathbf{x}) = \big(\frac{\partial}{\partial \mathbf{x}} L_{\mathbf{f}}^{n-1} p(\mathbf{x}), \mathbf{f}(\mathbf{x})\big)$, for $n > 0$,

where $(\cdot,\cdot)$ is the inner product of two vectors, i.e. $\big((a_1, \ldots, a_n), (b_1, \ldots, b_n)\big) = \sum_{i=1}^n a_i b_i$. Using Lie derivatives, (2) rewritten as

$$p(\mathbf{x}(t)) = L_{\mathbf{f}}^0 p(\mathbf{x}_0) + L_{\mathbf{f}}^1 p(\mathbf{x}_0) \cdot t + L_{\mathbf{f}}^2 p(\mathbf{x}_0) \cdot \frac{t^2}{2!} + \cdots + L_{\mathbf{f}}^i p(\mathbf{x}_0) \frac{t^i}{i!} + \cdots \tag{3}$$

Combining (1) and (3), our main result of continuous invariant generation (in the simplest case) can be stated as follows.

Theorem 2 (Necessary and Sufficient Criterion for CIs [21]). *Given a system $(D, \mathbf{f})$ with $D \mathrel{\widehat{=}} h(\mathbf{x}) > 0$, it has a continuous invariant of the form $P \mathrel{\widehat{=}} p(\mathbf{x}) \geq 0$ if and only if $\forall \mathbf{x}.\big(p(\mathbf{x}) = 0 \wedge h(\mathbf{x}) > 0 \longrightarrow \psi(p, \mathbf{f})\big)$, where*

$$\psi(p, \mathbf{f}) \mathrel{\widehat{=}} \begin{array}{ll} & L_{\mathbf{f}}^1 p(\mathbf{x}) > 0 \\ \vee & L_{\mathbf{f}}^1 p(\mathbf{x}) = 0 \wedge L_{\mathbf{f}}^2 p(\mathbf{x}) > 0 \\ \vee & \cdots \\ \vee & L_{\mathbf{f}}^1 p(\mathbf{x}) = 0 \wedge \cdots \wedge L_{\mathbf{f}}^{N_{p,\mathbf{f}}-1} p(\mathbf{x}) = 0 \wedge L_{\mathbf{f}}^{N_{p,\mathbf{f}}} p(\mathbf{x}) > 0 \\ \vee & L_{\mathbf{f}}^1 p(\mathbf{x}) = 0 \wedge \cdots \wedge L_{\mathbf{f}}^{N_{p,\mathbf{f}}-1} p(\mathbf{x}) = 0 \wedge L_{\mathbf{f}}^{N_{p,\mathbf{f}}} p(\mathbf{x}) = 0 \end{array}$$

with $N_{p,\mathbf{f}} \in \mathbb{N}$ computed from p and $\mathbf{f}$.

Proof. Please refer to [21]. □

Intuitively, Theorem 2 means that on the boundary of P, up to the $N_{p,\mathbf{f}}$-th order, the first non-zero higher order Lie derivative of p w.r.t $\mathbf{f}$ is non-negative.

The above theorem can be generalized for parametric polynomials $p(\mathbf{u}, \mathbf{x})$, thus enabling us to use polynomial templates and QE to automatically discover CIs. Such a method for CI generation is relatively complete, that is, if there exists a CI in the form of the predefined template, then we are able to find one.

Example 3. Consider the system $(\mathbb{R}^2, \mathbf{f})$ from [39] with $\mathbf{f} \mathrel{\widehat{=}} (\dot{x} = 1 - y, \dot{y} = x)$, which has a continuous invariant $p \geq 0$ with $p \mathrel{\widehat{=}} -(-x^2 - y^2 + 2y)^2$, defining the circumference of a circle.

In [39], sound and complete inference rules are given for invariants that are linear, quadratic, smooth or convex. However, it was pointed out in [39] that all these rules failed to prove the invariance property of $p \geq 0$, as p is not linear or quadratic, nor is it smooth or convex. Furthermore, by a simple computation we get $L_{\mathbf{f}}^k p \equiv 0$ for all $k \geq 1$, so the sound but incomplete rule in [39,38] which involves only strict inequalities over finite-order Lie derivatives is also inapplicable. However, from $L_{\mathbf{f}}^1 p \equiv 0$ we get $N_{p,\mathbf{f}} = 0$, and then according to Theorem 2, $p \geq 0$ can be verified since $\forall x \forall y. \big(-(-x^2 - y^2 + 2y)^2 = 0 \longrightarrow \mathsf{true}\big)$ holds trivially.

Although the rule in [26] can also be used to check the invariant $p \geq 0$, generally it only works on very restricted invariants. Even for linear systems like $(\mathbb{R}, \dot{x} = x)$, it cannot prove the invariant $x \geq 0$ because $\forall x. x \geq 0$ is obviously false, while our approach requires $\forall x.(x = 0 \rightarrow \mathsf{true})$ which is trivially true.

The above examples show the generality and flexibility of our approach, using which it is possible to generate CIs in many general cases, and hence gives more possibility to synthesize a controller based on our understandings of the kind of controllers that can be synthesized using methods in [38,40,37].

3.4 Heuristics for Predefining Templates

The key steps of the qualitative analysis used in [17] are as follows.

1. The evolution behavior (increasing or decreasing) of continuous variables in each mode is inferred from the differential equations (using first or second order derivatives);
2. *control critical* modes, at which the maximal (or minimal) value of a continuous variable is achieved, can be identified;
3. the safety requirement is imposed to obtain constraints on guards of transitions leading to control critical modes, and
4. then this information is propagated to other modes.

Next, we illustrate how such an analysis helps in predefining templates for the running example.

Example 4 (Nuclear Reactor Temperature Control). Our goal is to synthesize a switching controller for the system in Example 1 with the global safety requirement that the temperature of the core lies between 510 and 550, i.e. $S_i \mathrel{\widehat{=}} 510 \leq x \leq 550$ for $i = 1, 2, 3, 4$.

1) **Refine domains.** Using the safety requirement, domains D_i for $i = 1, 2, 3, 4$ are refined by $D_i^s \widehat{=} D_i \cap S_i$, e.g. $D_1^s \widehat{=} p = 0 \wedge 510 \leq x \leq 550$.
2) **Infer continuous evolutions.** Let $l_1 \widehat{=} x/10 - 6p - 50 = 0$ be the *zero-level* set of $\dot{x}$ and check how x and p evolve in each mode. For example, in D_2^s, $\dot{x} > 0$ on the left of l_1 and $\dot{x} < 0$ on the right; since p increases from 0 to 1, x first increases then decreases and achieves maximal value when crossing l_1.
3) **Identify critical control modes.** By 2), q_2 and q_4 are critical control modes at which the evolution direction of x changes.
4) **Generate control points.** By 3), we can get a control point $E(5/6, 550)$ at q_2 by taking the intersection of l_1 and the safety upper bound $x = 550$; and $F(1/6, 510)$ can be obtained similarly at q_4.
5) **Propagate control points.** E is backward propagated to $A(0, a)$ using trajectory $\widehat{AE}$ at q_2, and then to $C(1, c)$ using trajectory $\widehat{CA}$ at q_4; similarly, by propagating F we get D and B. (See Fig. 3.)
6) **Construct templates.** For brevity, we only show how to construct D_2'. Intuitively, $p = 0$, $p = 1$, $\widehat{AE}$ and $\widehat{BD}$ forms the boundaries of D_2'. In order to get a semi-algebraic template, we need to fit $\widehat{AE}$ and $\widehat{BD}$ by polynomials using points A, E and B, D respectively. By 2), $\widehat{AE}$ has only one extreme point E in D_2^s and is tangential to $x = 550$ at E. The simplest algebraic curve that can exhibit a shape similar to $\widehat{AE}$ is the parabola through A, E opening downward with $l_2 \widehat{=} p = \frac{5}{6}$ the axis of symmetry. Therefore to minimize the degree of terms appearing in templates, we do not resort to polynomials with degree greater than 3. This parabola can be computed using the coordinates of A, E as: $x - 550 - \frac{36}{25}(a - 550)(p - \frac{5}{6})^2 = 0$.

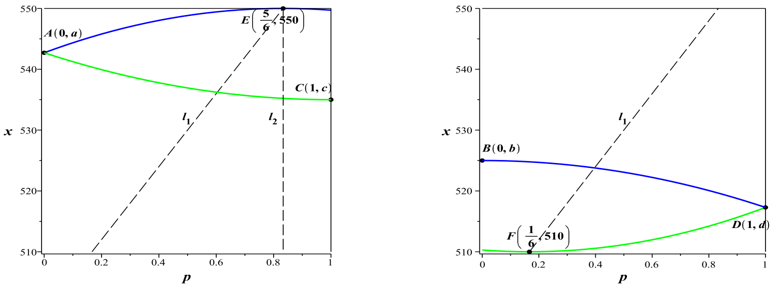

Fig. 3. Control points propagation.

Through the above analysis, we generate the following templates:

- $D_1' \widehat{=}\ p = 0 \wedge 510 \leq x \leq a$;
- $D_2' \widehat{=}\ 0 \leq p \leq 1 \wedge x - b \geq p(d - b) \wedge x - 550 - \frac{36}{25}(a - 550)(p - \frac{5}{6})^2 \leq 0$;
- $D_3' \widehat{=}\ p = 1 \wedge d \leq x \leq 550$;
- $D_4' \widehat{=}\ 0 \leq p \leq 1 \wedge x - a \leq p(c - a) \wedge x - 510 - \frac{36}{25}(d - 510)(p - \frac{1}{6})^2 \geq 0$,

in which a, b, c, d are parameters. Note that without qualitative analysis, a single generic *quadratic* polynomial over p and x would require $\binom{2+2}{2} = 6$ parameters.

The above heuristics works well on planar systems and can also be applied to three-dimensional systems. We are further generalizing the heuristics to cover a wider class of hybrid automata.

Based on the framework presented in Section 3.2, we show below how to synthesize a switching controller for the system in Example 4 step by step.

Example 5 (Nuclear Reactor Temperature Control Contd.).

(s1) The four invariant templates are defined in Section 3.4.

(s2) The four guards are refined by setting $G'_{ij} \mathrel{\widehat{=}} G_{ij} \cap D'_j$:
- $G'_{12} \mathrel{\widehat{=}} p = 0 \wedge b \le x \le a\,;$
- $G'_{23} \mathrel{\widehat{=}} p = 1 \wedge d \le x \le 550\,;$
- $G'_{34} \mathrel{\widehat{=}} p = 1 \wedge d \le x \le c\,;$
- $G'_{41} \mathrel{\widehat{=}} p = 0 \wedge 510 \le x \le a\,.$

(s3) Using D'_i and G'_{ij} we can derive the synthesis condition, which is a first-order polynomial formula in the form of $\phi \mathrel{\widehat{=}} \forall x \forall p. \varphi(a, b, c, d, x, p)$. We do not include ϕ here due to its big size.

(s4) By applying QE to ϕ we get the following solution to the parameters:

$$a = \frac{6575}{12} \wedge b = \frac{4135}{8} \wedge c = \frac{4345}{8} \wedge d = \frac{6145}{12}\ . \tag{4}$$

(s5) Instantiate D'_i and G'_{ij} by (4). It is obvious that all D'_i are nonempty closed sets. According to Theorem 1, we get a safe switching controller for the nuclear reactor system. The left picture in Fig. 4 is an illustration of D'_2.

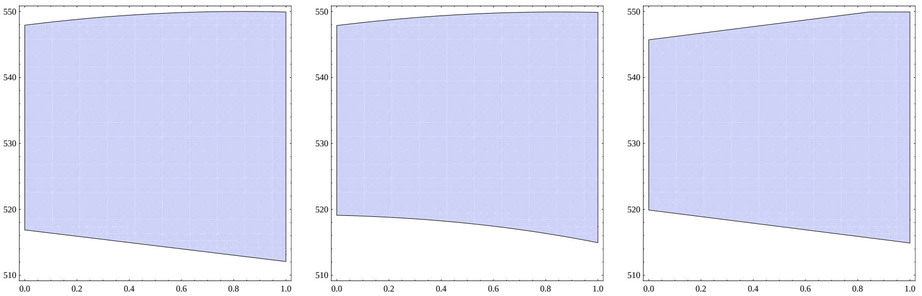

Fig. 4. Shape of synthesized continuous invariants.

In [17], an upper bound $x = 547.97$ for G_{12} and a lower bound $x = 512.03$ for G_{34} are obtained by solving the differential equations at mode q_2 and q_4 respectively. By (4), the corresponding bounds generated here are $x \le \frac{6575}{12} = 547.92$ and $x \ge \frac{6145}{12} = 512.08$.

As should be evident from the above results, in contrast to [17], where differential equations are solved to get closed-form solutions, we are able to get good approximate results without requiring closed-form solutions. This indicates that our approach should work well for hybrid automata where differential equations for modes need not have closed form solutions.

4 Synthesis by Generating CIs Numerically

The QE-based approach crucially depends upon quantifier elimination techniques. It is well known that the complexity of a general purpose QE method over the full theory of real-closed fields is *doubly exponential* in the number of variables [9]. Therefore it is desirable to develop heuristics to do QE more efficiently. As shown in Section 3.4, qualitative analysis helps in reducing the number of parameters in templates. Another possible way to address the issue of high computational cost is resorting to numerical methods. In this section, we will discuss the application of two such approaches to the nuclear reactor example.

4.1 The SOS Relaxation Approach

Let $\mathbb{R}[x_1, x_2, \ldots, x_n]$, or $\mathbb{R}[\mathbf{x}]$ for short, denote the polynomial ring over variables $x_1, x_2, \ldots, x_n$ with real coefficients. A *monomial* is a special polynomial in the form of $x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$ with $(\alpha_1, \alpha_2, \ldots, \alpha_n) \in \mathbb{N}^n$. Any polynomial $p(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ of degree d can be written as a linear combination of $\binom{n+d}{d}$ monomials, i.e.

$$p(\mathbf{x}) = \sum_{\alpha_1+\alpha_2+\cdots+\alpha_n \le d} c_{(\alpha_1,\alpha_2,\ldots,\alpha_n)} \cdot x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n} \quad .$$

A polynomial p is called an SOS (sum-of-squares) if there exist s polynomials $q_1, q_2, \ldots, q_s$ s.t.

$$p = \sum_{1 \le i \le s} q_i^2 \quad .$$

It is obvious that any SOS p is non-negative, i.e. $\forall \mathbf{x} \in \mathbb{R}^n .\, p(\mathbf{x}) \ge 0$.

The basic idea of SOS relaxation is as follows: to prove that a polynomial p is nonnegative, we can try to show that p can be decomposed into a sum of squares, a trivially sufficient condition for non-negativity (but generally not necessary); similarly, to prove $p \ge 0$ on the semi-algebraic set $q \ge 0$, it is sufficient to find two SOS r_1, r_2 such that $p = r_1 + r_2 \cdot q$.

SOS relaxation is attractive because the searching for SOS decomposition can be reduced to a semi-definite programming (SDP) problem according to the following equivalence [25]:

> A polynomial p of degree $2d$ is an SOS if and only if there exists a semi-definite matrix Q such that $p = \mathbf{q} \cdot Q \cdot \mathbf{q}^T$, where $\mathbf{q}$ is a $\binom{n+d}{d}$-dimensional row vector of monomials with degree $\le d$.

SDP is a convex programming that is solvable in *polynomial* time using numerical methods such as the interior point method [44]. Therefore the SOS computation is a tractable problem.

We now show how SOS can be related to CI generation. Let $p \geq 0$ be a parametric template defined for the system $(h > 0, \mathbf{f})$. By Theorem 2, a sufficient condition for $p \geq 0$ to be a CI of $(h > 0, \mathbf{f})$ is

$$\forall \mathbf{x}. \big(p(\mathbf{x}) = 0 \wedge h(\mathbf{x}) > 0 \longrightarrow L_{\mathbf{f}}^{1} p(\mathbf{x}) > 0\big) ,$$

which can be further strengthened to

$$\forall \mathbf{x}. \big(h(\mathbf{x}) > 0 \longrightarrow L_{\mathbf{f}}^{1} p(\mathbf{x}) > 0\big) . \tag{5}$$

Using SOS relaxation, a sufficient condition for (5) is

$$L_{\mathbf{f}}^{1} p = s_1 + s_2 \cdot h + \varepsilon \ \ , \tag{6}$$

where s_1, s_2 are SOS and ε is a positive constant. Solve (6) for the parameters appearing in p and then we can get a CI $p \geq 0$ of $(h > 0, \mathbf{f})$.

For the nuclear reactor example, we define two general *quartic* templates

$$p \geq 0 \wedge p \leq 1 \wedge \sum_{\alpha_1 + \alpha_2 \leq 4} c_{(\alpha_1, \alpha_2)} \cdot p^{\alpha_1} x^{\alpha_2} \leq 0$$

and

$$p \geq 0 \wedge p \leq 1 \wedge \sum_{\alpha_1 + \alpha_2 \leq 4} d_{(\alpha_1, \alpha_2)} \cdot p^{\alpha_1} x^{\alpha_2} \leq 0$$

for mode q_2 and q_4 respectively. Using the SOS relaxation techniques discussed above, the following refined domains are obtained:

- $D_1' \,\widehat{=}\, p = 0 \,\wedge\, 510 \leq x \leq 547.85\,;$
- $D_2' \,\widehat{=}\, 0 \leq p \leq 1 \,\wedge\, 34468.9 - 9941.89p + 1114.38p^2 + 67.3261p^3 + 0.925p^4 - 129.316x + 37.7294px - 4.9669p^2x - 0.1303p^3x + 0.1212x^2 - 0.0358px^2 + 0.0054p^2x^2 \leq 0\,;$
- $D_3' \,\widehat{=}\, p = 1 \,\wedge\, 512.09 \leq x \leq 550\,;$
- $D_4' \,\widehat{=}\, 0 \leq p \leq 1 \,\wedge\, 46082.8 + 8787.98p + 1473.0p^2 + 93.8933p^3 + 1.635p^4 - 174.456x - 34.1747px - 4.7397p^2x - 0.1829p^3x + 0.1649x^2 + 0.0332px^2 + 0.0037p^2x^2 \leq 0\,.$

The picture in the middle of Fig. 4 illustrates the synthesized D_2'.

4.2 The Template Polyhedra Approach

Polyhedral sets are a popular family of (positive) invariants of linear (continuous or discrete) systems [5]. A *convex polyhedron* in $\mathbb{R}^n$ can be represented using linear inequality constraints as $Q\mathbf{x} \leq \rho$, where $Q \in \mathbb{R}^{r \times n}$ is an $r \times n$ matrix, and $\mathbf{x} \in \mathbb{R}^{n \times 1}, \rho \in \mathbb{R}^{r \times 1}$ are column vectors.

Given a linear continuous dynamical system $\dot{\mathbf{x}} = A\mathbf{x}$ with $A \in \mathbb{R}^{n \times n}$, the following result about (positive) polyhedral invariant set is established in [7].

Proposition 1. *The polyhedron $Q\mathbf{x} \le \rho$ is a positive invariant set of $\dot{\mathbf{x}} = A\mathbf{x}$ if and only if there exists an essentially non-negative*[3] *matrix $H \in \mathbb{R}^{r\times r}$ satisfying $HQ = QA$ and $H\rho \le 0$.*

By simply applying the famous *Farkas' lemma* [11], we can generalize Proposition 1 and give a sufficient condition for polyhedral CIs of linear dynamics with open polyhedral domain.

Proposition 2. *Let $\mathbf{f} \mathrel{\widehat{=}} A\mathbf{x} + \mathbf{b}$ and $D \mathrel{\widehat{=}} \mathbf{c}\mathbf{x} < a$, where $a \in \mathbb{R}$, $\mathbf{b} \in \mathbb{R}^{n\times 1}$ is a column vector, and $\mathbf{c} \in \mathbb{R}^{1\times n}$ is a row vector. Then the polyhedron $Q\mathbf{x} \le \rho$ is a CI of the system $(D, \mathbf{f})$ if there exist essentially non-negative matrix $H \in \mathbb{R}^{r\times r}$, and non-negative column vectors $\eta \ge 0, \xi \ge 0, \lambda \ge 0$ in $\mathbb{R}^{r\times 1}$ such that*

(1) $HQ + \xi\mathbf{c} - \mathsf{diag}(\lambda)QA = 0$;
(2) $H\rho + \eta + \xi a + \mathsf{diag}(\lambda)Q\mathbf{b} = 0$;
(3) $\xi + \eta > 0$,

where $\mathsf{diag}(\lambda)$ denotes the $r \times r$ diagonal matrix with the main diagonal λ.

Proof. Please refer to Appendix B. □

Proposition 2 serves as the basis of automatic generation of polyhedral CIs for linear systems. To reduce the number of parameters in a polyhedral template, we propose the use of *template polyhedra*. The idea is to partly fix the shape of the invariant polyhedra by fixing the orientation of their facets. Formally, a template polyhedron is of the form $Q\mathbf{x} \le \rho$ where Q is fixed a priori and ρ is to be determined. Any instantiation of ρ from $\mathbb{R}^{r\times 1}$ produces a concrete polyhedron. In this paper, since the system is planar, we choose Q in such a way that its row vectors form a set of uniformly distributed directions on a unit circle, i.e.

$$\mathbf{q}_i = \big(\cos(\frac{i-1}{r}2\pi), \sin(\frac{i-1}{r}2\pi)\big)$$

for $1 \le i \le r$, where $\mathbf{q}_i$ denotes the i-th row of Q. It is easy to see that $Q\mathbf{x} \le \rho$ is just a rectangle when $r = 4$, and an octagon when $r = 8$.

In order to determine ρ, we have to solve the constraints derived from Proposition 2. Note that since both H and ρ are indeterminate, the constraint (2) becomes *bilinear*, making the problem NP-hard [43] to solve. It is however still more tractable using modern BMI (bilinear matrix inequality) solvers compared to QE. The details of applying numerical solvers will be discussed in the Conclusion part.

Using octagonal templates[4] for mode q_2 and q_4 in the nuclear reactor example, we obtain the following refined domains.

[3] A square matrix is *essentially non-negative* if all its entries are non-negative except for those on the diagonal. Besides, given a matrix M, in this paper the notation $M \ge 0$, $M > 0$ and $M = 0$ should be interpreted entry-wisely.

[4] To reduce the number of facets needed in the template, we scaled the variable x by a factor of 0.2, i.e. let $x = 5x'$, and rescaled the generated invariants by 5.

- $D'_1 \mathrel{\widehat{=}} p = 0 \wedge 510 \leq x \leq 545.50\,;$
- $D'_2 \mathrel{\widehat{=}} Q(p,x)^T \leq \rho_1$ with

  $$Q = \begin{pmatrix} 5.0000 & 3.5355 & 0.0000 & -3.5355 & -5.0000 & -3.5355 & -0.0000 & 3.5355 \\ 0.0000 & 0.7071 & 1.0000 & 0.7071 & 0.0000 & -0.7071 & -1.0000 & -0.7071 \end{pmatrix}^T$$

  and

  $$\rho_1 = \begin{pmatrix} 5.0000 & 392.4429 & 549.9276 & 385.8688 & 0.0000 & -367.6169 & -514.5244 & -360.2745 \end{pmatrix}^T\,;$$

- $D'_3 \mathrel{\widehat{=}} p = 1 \wedge 514.50 \leq x \leq 550\,;$
- $D'_4 \mathrel{\widehat{=}} Q(p,x)^T \leq \rho_2$ with the Q in D'_2 and

  $$\rho_2 = \begin{pmatrix} 5.0000 & 384.1267 & 545.5548 & 384.7484 & 0.0000 & -360.6299 & -510.1431 & -360.1948 \end{pmatrix}^T\,.$$

In Fig. 4, the picture on the right illustrates the synthesized D'_2.

5 Conclusion and Discussion

We have extended a template-based approach for synthesizing switching controllers for semi-algebraic hybrid systems by combining symbolic invariant generation methods using quantifier elimination with qualitative methods to determine the likely shape of invariants. We have also investigated the application of numerical methods to gain high level of scalability and automation. A summary comparison of the three proposed approaches, i.e. the QE-based, SOS-relaxation and template-polyhedra approaches, can be given in the following aspects.

- **Applicability:** the QE-based approach can be applied to any semi-algebraic system; SOS relaxation techniques can be applied to semi-algebraic systems for which SOS encoding is possible; the template-polyhedra approach is only applicable to linear systems.
- **Design of Templates:** the QE-based approach demands much heuristics in determining templates, while the other two need little human effort.
- **Relative Completeness:** only the QE-based approach is relative complete w.r.t. the predefined family of templates, but we believe that the template-polyhedra approach can be made relatively complete by improving Prop. 2.
- **Quality of Controllers:** the QE-based and SOS-relaxation approaches can generate arbitrary (non-convex) semi-algebraic invariants, while the template-polyhedra approach can only generate convex polyhedral invariants; for the nuclear reactor example, we can see from Fig. 4 that the QE-based approach produced larger refined domains and transition guards, but such superiority is not a necessity and relies greatly on the quality of heuristics.
- **Computational Cost:** for the QE-based approach, we have used the algebraic tools Redlog [10] and QEPCAD (the slfq function) [6] to perform QE; for the numerical approaches, we use the MATLAB optimization toolbox YALMIP [23,24] as a high-level modeling environment and the interfaced external solvers SeDuMi [36] and PENBMI [18] (the TOMLAB [13] version)

Table 1. Templates and time cost of three controller synthesis approaches.

Approach		QE-based	SOS-relaxation	template-polyhedra
Tool		Redlog + slfq	YALMIP + SeDuMi	YALMIP + PENBMI
Template	NR	quadratic, #PARMS = 4	generic quartic	8 facets
	TS	quadratic, #PARMS = 2	generic quartic	10/12 facets
Time (sec)	NR	12.663	1.969	0.578
	TS	7.092	1.609	1.437

to solve the underlying SDP and BMI problems respectively. Table 1 shows the time cost of three approaches applied to the nuclear reactor (NR) example as well as a thermostat (TS) example from [14]. All computations are done on a desktop with a 2.66 GHz CPU and 4 GB memory. We can see that for these two examples the QE-based approach is consistently more expensive in time compared to numerical approaches.

- **Soundness:** the QE-based approach is exact while the other two approaches suffer from numerical errors which would cause the synthesis of unsafe controllers. The justification for use of numerical methods is that verification is much easier than synthesis. For example, we have verified posteriorly and symbolically the controllers synthesized by both numerical approaches in this paper. We could also directly encode some tolerance of numerical errors into the synthesis constraints to increase robustness and reduce the risk of synthesizing bad controllers.

Our analysis of a nuclear reactor example suggests the effectiveness of all three proposed approaches. We are currently experimenting with these (and more other) methods on more complex examples. We believe that there exists no single method that can solve all the problems. A practical way is to select the most suitable one(s) for any specific problem.

Although the focus of this paper is on the switching controller synthesis problem subject to safety requirements, we plan to extend the proposed approach for reachability and/or optimality requirements as well, by incorporating our previous results on *asymptotic stability* analysis [22] and a case study in optimal control [46].

Acknowledgements. We thank Dr. Jiang Liu for his contribution to our previous joint work on invariant generation. We also thank Dr. Matthias Horbach and Mr. ThanhVu Nguyen for their valuable comments on our old drafts.

References

1. Alur, R.: Formal verification of hybrid systems. In: EMSOFT'11. pp. 273–278. ACM (2011)

2. Alur, R., Couroubetis, C., Henzinger, T., Ho, P.H.: Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. In: Hybrid Systems. LNCS, vol. 736, pp. 209–229. Springer (1993)
3. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The algorithmic analysis of hybrid systems. Theor. Comput. Sci. 138(1), 3–34 (1995)
4. Asarin, E., Bournez, O., Dang, T., Maler, O., Pnueli, A.: Effective synthesis of switching controllers for linear systems. Proc. of the IEEE 88(7), 1011–1025 (Jul 2000)
5. Blanchini, F.: Set invariance in control. Automatica 35(11), 1747–1767 (Nov 1999)
6. Brown, C.W.: QEPCAD B: A program for computing with semi-algebraic sets using CADs. SIGSAM Bulletin 37, 97–108 (2003)
7. Castelan, E., Hennet, J.: On invariant polyhedra of continuous-time linear systems. IEEE Trans. Autom. Control 38(11), 1680–1685 (nov 1993)
8. Cousot, P.: Proving program invariance and termination by parametric abstraction, Lagrangian relaxation and semidefinite programming. In: Cousot, R. (ed.) VMCAI'05, LNCS, vol. 3385, pp. 1–24. Springer (2005)
9. Davenport, J.H., Heintz, J.: Real quantifier elimination is doubly exponential. J. Symb. Comput. 5(1-2), 29–35 (1988)
10. Dolzmann, A., Seidl, A., Sturm, T.: Redlog User Manual (Nov 2006), `http://redlog.dolzmann.de/downloads/`, edition 3.1, for redlog Version 3.06 (reduce 3.8)
11. Gulwani, S., Tiwari, A.: Constraint-based approach for analysis of hybrid systems. In: CAV'08. LNCS, vol. 5123, pp. 190–203. Springer (2008)
12. Ho, P.H.: The algorithmic analysis of hybrid systems. Ph.D. thesis, Cornell University (1995)
13. Holmström, K., Göran, A.O., Edvall, M.M.: User's Guide for TOMLAB/PENOPT. Tomlab Optimization (Nov 2006), `http://tomopt.com/docs/TOMLAB_PENOPT.pdf`
14. Jha, S., Gulwani, S., Seshia, S.A., Tiwari, A.: Synthesizing switching logic for safety and dwell-time requirements. In: ICCPS'10. pp. 22–31. ACM (2010)
15. Kapur, D.: A quantifier-elimination based heuristic for automatically generating inductive assertions for programs. Journal of Systems Science and Complexity 19(3), 307–330 (2006)
16. Kapur, D.: Automatically Generating Loop Invariants Using Quantifier Elimination. Technical Report, Department of Computer Science, University of New Mexico, Albuquerque, USA. (Dec 2003)
17. Kapur, D., Shyamasundar, R.K.: Synthesizing controllers for hybrid systems. In: Maler, O. (ed.) Proc. HART'97. LNCS, vol. 1201, pp. 361–375. Springer (1997)
18. Kočvara, M., Stingl, M.: PENBMI User's Guide (Version 2.1). PENOPT GbR (Mar 2006), `http://www.penopt.com/doc/penbmi2_1.pdf`
19. Lin, W., Wu, M., Yang, Z., Zeng, Z.: Exact safety verification of hybrid systems using sums-of-squares representation. CoRR abs/1112.2328 (2011), `http://arxiv.org/abs/1112.2328`
20. Liu, J., Lv, J., Quan, Z., Zhan, N., Zhao, H., Zhou, C., Zou, L.: A calculus for hybrid CSP. In: APLAS'10. LNCS, vol. 6461, pp. 1–15. Springer (2010)
21. Liu, J., Zhan, N., Zhao, H.: Computing semi-algebraic invariants for polynomial dynamical systems. In: EMSOFT'11. pp. 97–106. ACM (2011)
22. Liu, J., Zhan, N., Zhao, H.: Automatically discovering relaxed Lyapunov functions for polynomial dynamical systems. Mathematics in Computer Science 6(4), 395–408 (2012)

23. Löfberg, J.: YALMIP : A toolbox for modeling and optimization in MATLAB. In: Proc. of the CACSD Conference. Taipei, Taiwan (2004), http://users.isy.liu.se/johanl/yalmip
24. Löfberg, J.: Pre- and post-processing sum-of-squares programs in practice. IEEE Trans. Autom. Control 54(5), 1007–1011 (2009)
25. Parrilo, P.A.: Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization. Ph.D. thesis, California Institute of Technology, Pasadena, CA (May 2000), http://thesis.library.caltech.edu/1647/
26. Platzer, A., Clarke, E.M.: Computing differential invariants of hybrid systems as fixedpoints. In: CAV'08. LNCS, vol. 5123, pp. 176–189. Springer (2008)
27. Platzer, A.: Differential dynamic logic for hybrid systems. J. Autom. Reasoning 41(2), 143–189 (2008)
28. Prajna, S., Jadbabaie, A.: Safety verification of hybrid systems using barrier certificates. In: HSCC'04. LNCS, vol. 2993, pp. 477–492. Springer (2004)
29. Prajna, S., Jadbabaie, A., Pappas, G.J.: A framework for worst-case and stochastic safety verification using barrier certificates. IEEE Trans. Autom. Control 52(8), 1415–1428 (Aug 2007)
30. Prajna, S., Papachristodoulou, A., Seiler, P., Parrilo, P.: SOSTOOLS and its control applications. In: Henrion, D., Garulli, A. (eds.) Positive Polynomials in Control, LNCIS, vol. 312, pp. 273–292. Springer (2005)
31. Sankaranarayanan, S., Sipma, H., Manna, Z.: Non-linear loop invariant generation using Gröbner bases. In: POPL'04 (2004)
32. Sankaranarayanan, S., Dang, T., Ivančić, F.: A policy iteration technique for time elapse over template polyhedra. In: Egerstedt, M., Mishra, B. (eds.) HSCC'08, LNCS, vol. 4981, pp. 654–657. Springer (2008)
33. Sankaranarayanan, S., Dang, T., Ivančić, F.: Symbolic model checking of hybrid systems using template polyhedra. In: Ramakrishnan, C., Rehof, J. (eds.) TACAS'08, LNCS, vol. 4963, pp. 188–202. Springer (2008)
34. Sankaranarayanan, S., Sipma, H., Manna, Z.: Scalable analysis of linear systems using mathematical programming. In: Cousot, R. (ed.) VMCAI'05, LNCS, vol. 3385, pp. 25–41. Springer (2005)
35. Sassi, M.A.B., Girard, A.: Computation of polytopic invariants for polynomial dynamical systems using linear programming. Automatica 48(12), 3114–3121 (2012)
36. Sturm, J.F.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optimization Methods and Software 11-12, 625–653 (1999)
37. Sturm, T., Tiwari, A.: Verification and synthesis using real quantifier elimination. In: ISSAC'11. pp. 329–336. ACM (2011)
38. Taly, A., Gulwani, S., Tiwari, A.: Synthesizing switching logic using constraint solving. International Journal on Software Tools for Technology Transfer 13, 519–535 (2011)
39. Taly, A., Tiwari, A.: Deductive verification of continuous dynamical systems. In: FSTTCS'09. LIPIcs, vol. 4, pp. 383–394 (2009)
40. Taly, A., Tiwari, A.: Switching logic synthesis for reachability. In: EMSOFT'10. pp. 19–28. ACM (2010)
41. Tarski, A.: A Decision Method for Elementary Algebra and Geometry. University of California Press, Berkeley (May 1951)
42. Tomlin, C.J., Lygeros, J., Sastry, S.S.: A game theoretic approach to controller design for hybrid systems. Proc. of the IEEE 88(7), 949–970 (Jul 2000)
43. VanAntwerp, J.G., Braatz, R.D.: A tutorial on linear and bilinear matrix inequalities. Journal of Process Control 10(4), 363–385 (2000)

44. Vandenberghe, L., Boyd, S.: Semidefinite programming. SIAM Review 38(1), 49–95 (1996)
45. Yang, Z., Wu, M., Lin, W.: Exact safety verification of hybrid systems based on bilinear SOS representation. CoRR abs/1201.4219 (2012), http://arxiv.org/abs/1201.4219
46. Zhao, H., Zhan, N., Kapur, D., Larsen, K.G.: A "hybrid" approach for synthesizing optimal controllers of hybrid systems: A case study of the oil pump industrial example. In: Giannakopoulou, D., Méry, D. (eds.) FM'12, LNCS, vol. 7436, pp. 471–485. Springer (2012)

A Proof of Theorem 1

We need the following definitions [42] to prove Theorem 1.

Definition 3 (Hybrid Time Set). *A hybrid time set is a sequence of intervals $\tau = \{I_i\}_{i=0}^N$ (N can be ∞) such that:*

- *$I_i = [\tau_i, \tau_i']$ with $\tau_i \leq \tau_i' = \tau_{i+1}$ for all $i < N$;*
- *if $N < \infty$, then $I_N = [\tau_N, \tau_N'\rangle$ is a right-closed or right-open nonempty interval (τ_N' may be ∞);*
- *$\tau_0 = 0$.*

Given a hybrid time set, let $\langle\tau\rangle = N$ and $\|\tau\| = \sum_{i=0}^N (\tau_i' - \tau_i)$.

Definition 4 (Hybrid Trajectory). *A hybrid trajectory of $\mathcal{H}$ starting from an initial point $(q_0, \mathbf{x}_0) \in D_{\mathcal{H}}$ is a triple $\omega = (\tau, \alpha, \beta)$, where $\tau = \{I_i\}_{i=0}^N$ is a hybrid time set, and $\alpha = \{\alpha_i : I_i \to Q\}_{i=0}^N$ and $\beta = \{\beta_i : I_i \to \mathbb{R}^n\}_{i=0}^N$ are two sequences of functions satisfying:*

1. *Initial condition: $\alpha_0[0] = q_0$ and $\beta_0[0] = \mathbf{x}_0$;*
2. *Discrete transition: for all $i < \langle\tau\rangle$, $e = \big(\alpha_i(\tau_i'), \alpha_{i+1}(\tau_{i+1})\big) \in E$, $\beta_i(\tau_i') \in G_e$ and $\beta_{i+1}(\tau_{i+1}) = \beta_i(\tau_i')$;*
3. *Continuous evolution: for all $i \leq \langle\tau\rangle$ with $\tau_i < \tau_i'$, if $q = \alpha_i(\tau_i)$, then*
   (1) for all $t \in I_i$, $\alpha_i(t) = q$,
   (2) $\beta_i(t)$ is the solution to the differential equation $\dot{\mathbf{x}} = \mathbf{f}_q(\mathbf{x})$ over I_i starting from $\beta_i(\tau_i)$, and
   (3) for all $t \in [\tau_i, \tau_i')$, $\beta_i(t) \in D_q$.

Proof of Theorem 1

Proof. We prove that the three requirements in Problem 1 are satisfied by $\mathcal{H}'$.

(r1) By (c1), we get $D_q' \subseteq D_q$ for all $q \in Q$; by the definition of G_e', $G_e' \subseteq G_e$ for all $e \in E$.

(r2) Suppose $\omega = (\tau, \alpha, \beta)$ is a hybrid trajectory starting from $(q_0, \mathbf{x}_0) \in D_{\mathcal{H}'}$. We prove

$$\forall i \leq \langle\tau\rangle \, \forall t \in I_i. \, \beta_i(t) \in S_{\alpha_i(t)} \tag{7}$$

by induction on $\langle\tau\rangle$.

If $\langle\tau\rangle = 0$, then $I_0 = [0, T_0\rangle$ is a right-open or right-closed interval for some $T_0 \geq 0$. If $T_0 = 0$ then $I_0 = \{0\}$. By (c1) and condition 1 of Definition 4 we have $\beta_0(0) = \mathbf{x}_0 \in D_{q_0}' \subseteq S_{q_0} = S_{\alpha_0(0)}$. If $T_0 > 0$, by condition 1 and 3 of Definition 4 as well as (c1), we have for all $t \in [0, T_0)$, $\beta_0(t) \in D_{q_0}' \subseteq S_{q_0} = S_{\alpha_0(t)}$. If $I_0 \neq [0, T_0)$, i.e. $I_0 = [0, T_0]$, by noticing that D_{q_0}' is a closed set and $\beta_0(t)$ is continuous over I_0, we get $\beta_0(T_0) \in D_{q_0}' \subseteq S_{q_0} = S_{\alpha_0(T_0)}$. Thus we have proved in all cases, $\forall t \in I_0. \, \beta_0(t) \in S_{\alpha_0(t)}$.

Assume (7) holds for $\langle\tau\rangle = k \geq 0$. When $\langle\tau\rangle = k+1$, by assumption, for all $i \leq k$ and all $t \in I_i$ we have $\beta_i(t) \in S_{\alpha_i(t)}$. By condition 2 in Definition

4, there exists $e = (q_k, q_{k+1}) \in E$ such that $\alpha_k(\tau'_k) = q_k$, $\alpha_{k+1}(\tau_{k+1}) = q_{k+1}$ and $\beta_{k+1}(\tau_{k+1}) = \beta_k(\tau'_k) \in G'_e \subseteq D'_{q_{k+1}}$. Consider the hybrid trajectory starting from $\big(q_{k+1}, \beta_{k+1}(\tau_{k+1})\big) \in D_{\mathcal{H}'}$. By applying the same analysis we do for case $\langle\tau\rangle = 0$, we can get $\beta_{k+1}(t) \in S_{\alpha_{k+1}(t)}$ for all $t \in I_{k+1}$. Thus we have proved that (7) holds for $\langle\tau\rangle = k+1$. Then by induction, (7) holds for all $\langle\tau\rangle$.

(r3) Given $(q_0, \mathbf{x}_0) \in D_{\mathcal{H}'}$ (hence $D'_{q_0} \neq \emptyset$), we will construct a non-blocking hybrid trajectory starting from $(q_0, \mathbf{x}_0)$.

Suppose $\mathbf{x}(t)$ is the continuous evolution defined by $\mathbf{f}_{q_0}$ starting from $\mathbf{x}_0$. Let $T_{\max}$ be the maximal positive T satisfying

$$\mathbf{x}(t) \in D'_{q_0} \cap H_{q_0} \text{ for all } t \in [0, T)\,, \tag{8}$$

if such T exists, and $T_{\max} = 0$ otherwise.

If $T_{\max} = \infty$, then we already get an infinite hybrid trajectory starting from $(q_0, \mathbf{x}_0)$.

If $T_{\max} < \infty$, then by the completeness of $\mathbf{f}_{q_0}$, $\mathbf{x}(t)$ must exist on $[0, T_{\max} + \varepsilon]$ for some $\varepsilon > 0$. We assert that

$$\mathbf{x}(T_{\max}) \in (H_{q_0})^c = \bigcup_{e=(q_0,q')\in E} G'_e \ . \tag{9}$$

If not, i.e. $\mathbf{x}(T_{\max}) \in H_{q_0}$, then there exists $0 < \varepsilon' < \varepsilon$ s.t. $\mathbf{x}(t) \in H_{q_0}$ on $[0, T_{\max} + \varepsilon']$, because by assumption H_{q_0} is an open set. Then by (c2) and the definition of continuous invariant, we get $\mathbf{x}(t) \in D'_{q_0} \cap H_{q_0}$ on $[0, T_{\max} + \varepsilon']$, so $T_{\max}$ could not be maximal. Therefore (9) holds. Then there exists $e = (q_0, q') \in E$ such that $\mathbf{x}(T_{\max}) \in G'_e \subseteq D'_{q'}$, so we can make a discrete jump from q_0 to q' and extend the hybrid trajectory by continuous evolution at q'.

Such extension either ends with a trajectory satisfying $\|\tau\| = \infty$ or goes on forever resulting $\langle\tau\rangle = \infty$. □

B Proof of Proposition 2

To prove Proposition 2, we first introduce the famous *Farkas' lemma* [11] in the theory of linear programming.

Lemma 1 (Farkas' Lemma). *For linear formulas $p_j, r_k \in \mathbb{R}[\mathbf{x}]$, the formula $\bigwedge_{j\in J} p_j > 0 \wedge \bigwedge_{k\in K} r_k \geq 0$ is unsatisfiable (over the reals) if and only if there exist non-negative constants μ, μ_j $(j \in J)$ and ν_k $(k \in K)$ such that*

$$\mu + \sum_{j\in J} \mu_j p_j + \sum_{k\in K} \nu_k r_k = 0$$

and at least one of μ_j, μ is strictly positive.

Proof of Proposition 2

Proof. Let $\mathbf{q}_i$ denote the i-th row vector of the matrix Q and ρ_i denote the i-th entry of column vector ρ. Then $\mathbf{q}_i\mathbf{x} = \rho_i$ stands for the i-th facet of the polyhedron $Q\mathbf{x} \leq \rho$. By a generalization of Theorem 2 (see [21] for the detail), $Q\mathbf{x} \leq \rho$ is a CI of $(\mathbf{cx} < a, A\mathbf{x} + \mathbf{b})$ if for all $1 \leq i \leq r$, the following implication holds:

$$Q\mathbf{x} \leq \rho \wedge \mathbf{q}_i\mathbf{x} = \rho_i \wedge \mathbf{cx} < a \Longrightarrow \mathbf{q}_i(A\mathbf{x} + \mathbf{b}) < 0 \ ,$$

which is equivalent to

$$-Q\mathbf{x} + \rho \geq 0 \wedge \mathbf{q}_i\mathbf{x} - \rho_i \geq 0 \wedge \mathbf{q}_i(A\mathbf{x} + \mathbf{b}) \geq 0 \wedge -\mathbf{cx} + a > 0 \tag{10}$$

is unsatisfiable.

By Lemma 1, the unsatisfiability of (10) is equivalent to the existence of constant γ_i^i, and non-negative constants $\gamma_i^1, \gamma_i^2, \ldots, \gamma_i^{i-1}, \gamma_i^{i+1}, \ldots \gamma_i^r$, η_i, λ_i, ξ_i such that

$$\sum_{1 \leq i \leq r} \gamma_i(-\mathbf{q}_i\mathbf{x} + \rho_i) + \lambda_i\mathbf{q}_i(A\mathbf{x} + \mathbf{b}) + \xi_i(-\mathbf{cx} + a) + \eta_i = 0 \tag{11}$$

and $\xi_i + \eta_i > 0$. By equating the coefficients of the left side of (11) to 0, we get

(1) $\sum_{i=1}^{r}(-\gamma_i\mathbf{q}_i) + \lambda_i\mathbf{q}_iA - \xi_i\mathbf{c} = 0\,;$
(2) $\sum_{i=1}^{r}\gamma_i\rho_i + \lambda_i\mathbf{q}_i\mathbf{b} + \xi_i a + \eta_i = 0\,;$ and
(3) $\xi_i + \eta_i > 0\,,$

the matrix form of which corresponds to the three conditions in Proposition 2.

□